PTO/SB/21 (08-08)
Approved for use through 09/30/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM (to be used for all correspondence after initial filing)			
		Application Number	10/815,294
		Filing Date	Apr 1, 2004
		First Named Inventor	Barry, Edwin Franklin
		Art Unit	2188
		Examiner Name	Doan, Duc T.
Total Number of Pages in This Submission		Attorney Docket Number	800.0015 (A01455)

ENCLOSURES *(Check all that apply)*

- [ ] Fee Transmittal Form
- [ ] Fee Attached
- [ ] Amendment / Reply
- [ ] After Final
- [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
- [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
- [ ] Landscape Table on CD
- [ ] After Allowance communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [x] Appeal Communication to TC (Appeal Notice, Brief, Reply Brief)
- [ ] Proprietary Information
- [ ] Status Letter
- [ ] Other Enclosure(s) (please identify below):

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Priest & Goldstein, PLLC		
Signature	Peter H. Priest		
Printed name	Peter H. Priest		
Date	October 20, 2008	Reg. No.	30210

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Signature	Marianna Tortorelli		
Typed or printed name	Marianna Tortorelli	Date	October 20, 2008

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RECEIVED
CENTRAL FAX CENTER
OCT 20 2008

800.0015
A01455

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	Barry
For	:	Methods and Apparatus for Address Translation Functions
Serial No.	:	10/815,294
Filed	:	04/01/2004
Group	:	2188
Examiner	:	Doan, Duc T.

Durham, North Carolina
October 20, 2008

MAIL STOP APPEAL BRIEF – PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S REPLY BRIEF

Sir:

The present reply brief responds to points raised by the Examiner's Answer regarding claims 1, 3, 4, 7, 11, 18, and 19. More specifically, it addresses the Answer's technically incorrect analysis and application of Dowling, Intel, Nair, Choquette, and Pechanek. As discussed below, the Examiner's Answer suggests that the relied upon references meet the language of the claims, but as will be shown, the language of the claims is clear and the references do not meet the requirements of the claim language.

Regarding claim 1, at page 17, line 22 – page 18, line 2, the Examiner's Answer incorrectly seeks to compare the present specification's Fig 3 in which Rt 320 represents the address register as being similar to Dowling's AR0-ARn. Fig. 3 of the present application illustrates a block load instruction having a register file target Rt address 320 which is a beginning operand address of a block of operands located in a register file, as described at page 11, lines 10-13 of the present specification. Rt 320 is similar to Rt 142 of Fig. 1 representing the Rt field of an instruction stored in an instruction register 108 as described at page 7, lines 11-15 of the present specification. Rt 320 is not an address register but rather is a field of an instruction which is used to address an operand in a register file. By contrast, Dowling's Ar0-ARn are address registers and are not part of an instruction or part of an instruction register.

At page 19, lines 6-10, the Examiner's Answer states the "instant application discloses an address translation scheme to translate operand address of two types a) the operand field of an instruction is an address/logical number **that points to a register of registers**. This register number is regarded in the instant application as indirect type address and b) the operand field of an instruction refers **directly to address/pointer to memory location.**" Examiner's emphasis.

The Examiner's Answer is incorrect in this reading of the present application. The type a indirect type address is not a valid addressing type with regard to claims 1 and 16. In particular, the Examiner's Answer at page 19, lines 8 and 9 states that a "register number is regarded in the instant application as indirect type address". Indirect addressing is a commonly used term that concerns a mechanism for identifying an operand in storage by the operand's storage address that is stored in an intermediate register, also referred to as an address register. When an instruction using indirect addressing is executed, a bit field in the instruction points to the address register whose contents are then utilized as an address to access the operand from memory, such that, the

final memory address is indirectly specified by the bit field in the instruction. The present specification does not claim indirect addressing with regard to claims 1 and 16, but rather uses direct addressing and advantageously describes a translation unit for "directly translating the operand address bit field received as input" from the instruction register as claimed in claim 1. Dowling has no recognition of the problems advantageously solved by the instant invention as claimed in claim 1.

At page 19, lines 16-19, the Examiner's Answer states that "Intel teaches that typically the operand address field of an instruction can be several variations, including type b address of the instant disclosure. In Intel the displacement value points to a memory location. Therefore the Intel's displacement value is type b operand address as claimed". As described in the Appeal Brief filed June 16, 2008 at page 17, lines 14-23, Intel's "disp16 is not a direct addressing format but, rather, is used in the generation of an effective address by adding the disp16 value to an index value."

At page 21, lines 1-6, the Examiner's Answer states that "Dowling's manipulation of operand address bits AR0-ARn cover the claim's operand address translation (the AR0-ARn in the register set 102 stores the operand address, col. 1, lines 20-51; col. 9 lines 1-15, col. 10 lines 14-18, by manipulating the address bit by the functions of a logic device AA212, the operand address translation is achieved". Dowling's address registers AR0-ARn in register set 102 are not an "operand address bit field of the instruction" that is stored in an "instruction register" as claimed in claim 1. Dowling shows no path for the claimed "operand address bit field of the instruction" stored in an "instruction register" to Dowling's programmable AAU 212 and provides no motivation that such a path would provide any value. To the contrary, Dowling explicitly states that the programmable AAU 301 of Fig. 3A may be used "as the programmable

AAU 212 in the processor 200 shown in Fig. 2" which "allows a programmer to programmably permute the bits in the address registers AR0-ARn in the register set 102." Dowling, Figs. 2, 3A, and 3C and col. 10, lines 14-18. As is generally known in the art, an address register is not an instruction register, and a value stored in an address register is not the same as a value stored in an instruction register. An address register stores an address value used to indirectly address memory as described above. An address register and the value stored therein is used differently function than an instruction register and the value stored therein. Dowling merely extends the capabilities of an address arithmetic unit (AAU) by programmably modifying address values stored in address registers. Dowling does not teach and does not make obvious an "instruction register for receiving an instruction encoded with an operand address in an operand address bit field of the instruction and control information indicating the operand address is to be translated as part of the instruction's execution" as claimed in claim 1.

Regarding claim 4, the Examiner's Answer cites new art as "evidentiary reference" in his argument for rejecting claim 4. Any reliance upon new art at this point in time is inappropriate and should be withdrawn. Further, this new art does not meet the language of claim 4. In particular, claim 4 contains all the limitations of the base claim 1, thus the new art is clearly not commensurate with claim 4.

With regard to claims 3 and 19, the Examiner's Answer at page 23, lines 18 and 19 incorrectly suggests "that the matrix as recited in claim 3 grouped a group and b group above, merely indicates the basic bitwise operations as taught by Nair". However, Nair's specific bitiwise operations, such as AND/NAND/OR/NOR/XOR and similar logical operations such as shown in Nair table 2, col. 13 are individual logical operations, such as MSHRA used for a shift right arithmetic operation, MAND used for a matrix logical AND operation, and the like. Such

specific bitwise operations do not describe a set of equations which provide a unique combination of logical operations on each bit of "the operand address bit field input" as claimed in claims 3 and 19.

With regard to claim 18, the Examiner's Answer states at page 24, lines 11 and 12 that the phrase "address translation memory" is not found anywhere in the specification. However, the phrase "address translation memory" is found in the original claim 11 which is part of the specification. As claimed in the original claim 11, an address translation memory device comprises "a first read address input; a storage device having data addressable at addressable locations, a second read address input for selecting data from the storage device during read operations, and a data output port; and an address translation unit for accessing the storage device in a translation pattern, the address translation unit translating the first read address input in accordance with the translation pattern, to the storage device read address input for reading data from the storage device at a translated address during a read operation." Such an address translation memory is shown in Fig. 8A's storage unit 810 which has a first read address input 815, a storage device 835 having data accessible at addressable locations in a storage array 845 and a second read address input A2', A1', and A0', and an address translation unit 830 for translating the first read address input 815 to the storage device second read address input A2', A1', and A0'.

The Examiner's Answer also states at page 26, lines 14-19 that "Fig. 2D clearly shows all address inputs must go through the address translation function 243, these address inputs are translated according to the load translation parameters 248 that govern how the addresses access data from the storage unit 238. In other words, Fig. 2D behaves the same as Fig. 8A, and all address bits are being translated." However, Fig. 2D includes "a control input 252 from a decode

and control unit, such as decode and control unit 118 of Fig. 1" as described by the present specification at page 9, lines 20 and 21. Also, the "translation parameter control unit 246 may also receive decode and control information 252 indicating, for example, whether an instruction is to use or not use the address translation function" as described at page 10, lines 13-15. With regard to Fig. 8A, the storage subsystem 800 illustrates "aspects of a read port, which operates differently than the storage subsystem 238 shown in Fig. 2D. ... In Fig. 2D subsystem 230, the address translation occurs only in operation for instructions that specify the address translation function. Other instructions, which don't specify an address translation function, use the register file or memory unit normally as sequentially addressed storage. In the storage subsystem 800 of Fig. 8A, all address inputs 815 are translated according to the translation settings 820 that govern how the addresses access data from the storage unit 835." Specification, page 21, line 13 – page 22, line 1. The storage subsystem 800 of Fig. 8A does not have an input similar to the control input 252 of Fig, 2D. Thus, claim 18 does not claim a rearrangement of old elements providing an already known function.

The Examiner's Answer at page 27, lines 11-13 states that "Dowling's data memory 102 have the data correspond to **the address registers AR0 to ARn. These address registers contains addresses that used for address translation, and thus belong to 'the address translation memory".** Examiner's emphasis. The Examiner's Answer also states at page 28, lines 9-12 that "Dowling teaches, ... the registers 308 in the AAU that can be viewed as the address translation memory". The Examiner's Answer further states at page 28, lines 14-16, that "the cross bar logic translates input 302 to output 304 is part of the AAU which can be viewed as the address translation memory". However, Dowling's address registers AR0-ARn, the registers 308 in the AAU, and the cross bar logic do not meet the language of claim 18. Claim 18 recites

an "address translation method for translating a first address of a first data element in an address translation memory to a second address of a second data element in the address translation memory". As claimed, the first data element and the second data element are located in the address translation memory. If we consider the Examiner's Answer's position that the address registers 102, AAU 212 containing program register 2 (PR2) 308 cross bar logic 301 having input from address registers 302 and output to address registers 304 comprise an address translation memory, then Dowling does not translate an address of a data element in address translation memory as claimed. Rather, Dowling translates the contents of an address register AR0-ARn, but not the address that points to one of the address registers.

The Examiner's Answer at page 29, lines 4-6 further incorrectly states that "Appellant further argues 'Dowling translates an address before any read operation to the data memory 120 occurs'. In response, Appellant argument is not understood". Dowling shows numerous examples to support the translating of an address before any read operation to the data memory 120 occurs. For example, Dowling's auto update store operation begins with the processor performing "the following actions (assuming that AR0 is the default address register): (1) set the multiplexer 122 to select an address from the default address register (e.g. AR0), (2) put the data in AR1 on the data bus 112, (3) store the data on the data bus 112 in the data-memory 120 at the address specified by the default address register, (4) use the AAU 106 to increment the address in the default address register." Dowling, col. 8, lines 40-48. As is generally known in the art, the auto increment of an address is done while a previously stored address is used in the execution of an instruction. This approach is used to avoid any increase in the critical timing path delay for forming the incremented address for use by an instruction in execution. In the next execution of the instruction, the updated address is already stored for use. Dowling further

states at col. 9, lines 12-15, "the programmable AAU 212 can be programmed to provide automatic address indexing", meaning Dowling translates an address before any read operation to the data memory 120 occurs. As such, Dowling provides no support for "initiating a read operation to read a first data element at a first address during a read operation; translating the first address to the second address in accordance with the {s,e} bit specified translation pattern; and completing the read operation by reading the second data element at the second address" as claimed in claim 18.

Regarding claim 7, the Examiner's Answer at page 31, lines 3-5, states that "Dowling discloses that in an FFT operation, the first operation does not need operand address translation, and the subsequent operand addresses can be translated" citing Dowling col. 7, lines 1-15 for support. However, the cited text teaches nothing of the sort, but merely describes "bit reversal addressing modes that are commonly used to implement FFT algorithms. Bit reversal, loosely speaking, provides for reversing the order of the least significant bits in an address register. Given an 8-bit register containing the value 01001000b (the b suffix meaning binary), reversing the order of the lowest four bits in the register gives the value 01000001b." Nowhere does Dowling say that the first operation does not need operand address translation. Thus, Dowling does not cure the admitted deficiencies of Pechanek 6,173,389.

Regarding claim 11, the Examiner's Answer at page 32, lines 17-22 states that "**a second** read address input port **to the storage device 835** is not found anywhere in the Fig. 8A. Instead Fig. 8A clearly shows only single port to the storage device 835 which come from the same first read address input port with address input 815. In fact, the storage device 835 clearly shows as having a single read address input port A2' A1' A0'. Noted that the port A2' A1' A0' is external to storage device 835 and of course/inherently the port A2' A1' A0' is internal to the address

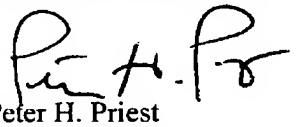
translation memory device." Examiner's emphasis. No further analysis or conclusion is provided by the Examiner's Answer.

In "the storage subsystem 800 of Fig. 8A, all address inputs 815 are translated according to the translation settings 820 that govern how the addresses access data from the storage unit 835." Present specification, page 21, line 22 – page 22, line 1. To be clear, exemplary reference numerals from the storage subsystem 800 of Fig. 8A are added to the corresponding elements of claim 11 below. As claimed in claim 11, the address translation memory device 810 comprises "a first read address input port (815) to the address translation memory device (810); storage device (835) located in the address translation memory device (810) having data accessible at addressable locations (845), a second read address input port (A2' A1' A0') internal to the address translation memory device (810)". Reference notation and underlining added for emphasis.

Conclusion

The claims on appeal should be promptly passed to issue.

Respectfully submitted,



Peter H. Priest
Reg. No. 30,210
Priest & Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713
(919) 806-1600

CLAIMS APPENDIX
(Claims Under Appeal)

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.